



```
import pandas as pd
from scipy import stats
from sklearn.preprocessing import MinMaxScaler
import numpy as np
df=pd.read_csv("/bmi.csv")
print(df)
```



	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
..	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

[500 rows x 4 columns]

```
df.dropna()
```



	Gender	Height	Weight	Index
0	Male	0.809353	0.587322	4
1	Male	0.926867	0.375390	2
2	Female	0.813400	0.581704	4
3	Female	0.884810	0.465951	3
4	Male	0.836314	0.548250	3
...	...	...	...	...
495	Female	0.178116	0.984010	5
496	Female	0.756127	0.654425	4
497	Female	0.021674	0.999765	5
498	Male	0.382762	0.923847	5
499	Male	0.604868	0.796326	5

500 rows x 4 columns

Start coding or [generate](#) with AI.

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
df[['Height','Weight']]=scaler.fit_transform(df[['Height','Weight']])
df.head(10)
```



	Gender	Height	Weight	Index
0	Male	0.247939	-0.309117	4
1	Male	1.164872	-0.587322	2
2	Female	0.920357	0.123647	4
3	Female	1.531645	-0.061823	3
4	Male	-1.280283	-1.391027	3
5	Male	1.164872	-0.061823	3
6	Male	-1.402541	-0.432764	5
7	Male	-0.974639	0.154559	5
8	Male	0.247939	-0.494587	3
9	Female	-0.057706	-0.092735	4

```
scaler=MinMaxScaler()  
df[['Height','Weight']]=scaler.fit_transform(df[['Height','Weight']])  
df.head(10)
```



	Gender	Height	Weight	Index
0	Male	0.576271	0.418182	4
1	Male	0.830508	0.336364	2
2	Female	0.762712	0.545455	4
3	Female	0.932203	0.490909	3
4	Male	0.152542	0.100000	3
5	Male	0.830508	0.490909	3
6	Male	0.118644	0.381818	5
7	Male	0.237288	0.554545	5
8	Male	0.576271	0.363636	3
9	Female	0.491525	0.481818	4

```
from sklearn.preprocessing import Normalizer  
scaler=Normalizer()  
df[['Height','Weight']]=scaler.fit_transform(df[['Height','Weight']])  
df
```



	Gender	Height	Weight	Index
0	Male	0.809353	0.587322	4
1	Male	0.926867	0.375390	2
2	Female	0.813400	0.581704	4
3	Female	0.884810	0.465951	3
4	Male	0.836314	0.548250	3
...	...	...	...	...
495	Female	0.178116	0.984010	5
496	Female	0.756127	0.654425	4
497	Female	0.021674	0.999765	5
498	Male	0.382762	0.923847	5
499	Male	0.604868	0.796326	5

500 rows × 4 columns

Double-click (or enter) to edit

```
from sklearn.preprocessing import Normalizer
scaler=Normalizer()
df[['Height','Weight']]=scaler.fit_transform(df[['Height','Weight']])
df
```



	Gender	Height	Weight	Index
0	Male	0.625687	-0.780074	4
1	Male	0.892924	-0.450208	2
2	Female	0.991096	0.133150	4
3	Female	0.999186	-0.040331	3
4	Male	-0.677211	-0.735789	3
...	...	...	...	...
495	Female	-0.642809	0.766027	5
496	Female	0.880037	0.474905	4
497	Female	-0.885715	0.464230	5
498	Male	-0.963237	-0.268652	5
499	Male	0.234966	0.972004	5

500 rows × 4 columns



```
df3=pd.read_csv("/bmi.csv")
from sklearn.preprocessing import MaxAbsScaler
scaler=MaxAbsScaler()
df3[['Height','Weight']]=scaler.fit_transform(df3[['Height','Weight']])
df3
```



	Gender	Height	Weight	Index
0	Male	0.874372	0.60000	4
1	Male	0.949749	0.54375	2
2	Female	0.929648	0.68750	4
3	Female	0.979899	0.65000	3
4	Male	0.748744	0.38125	3
...	...	...	...	...
495	Female	0.753769	0.95625	5
496	Female	0.924623	0.75625	4
497	Female	0.708543	0.85000	5
498	Male	0.753769	0.59375	5
499	Male	0.869347	0.81875	5

500 rows × 4 columns



Start coding or [generate](#) with AI.

```
df4=pd.read_csv("/bmi.csv")
from sklearn.preprocessing import RobustScaler
scaler=RobustScaler()
df4[['Height','Weight']]=scaler.fit_transform(df3[['Height','Weight']])
df4
```



	Gender	Height	Weight	Index
0	Male	0.125000	-0.178571	4
1	Male	0.660714	-0.339286	2
2	Female	0.517857	0.071429	4
3	Female	0.875000	-0.035714	3
4	Male	-0.767857	-0.803571	3
...	...	...	...	...
495	Female	-0.732143	0.839286	5
496	Female	0.482143	0.267857	4
497	Female	-1.053571	0.535714	5
498	Male	-0.732143	-0.196429	5
499	Male	0.089286	0.446429	5

500 rows × 4 columns



```
import pandas as pd
import numpy as np
import matplotlib
import seaborn as sns
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.feature_selection import RFE
from sklearn.linear_model import RidgeCV, LassoCV, Ridge, Lasso
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_regression
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import chi2
df=pd.read_csv("titanic_dataset.csv")
df.columns
```



```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
      'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
df.shape
```



```
(891, 12)
```

```
X=df.drop("Survived", axis=1)
y=df['Survived']
X
y
```



Survived	
0	0
1	1
2	1
3	1
4	0
...	...
886	0
887	1
888	0
889	1
890	0

891 rows × 1 columns

dtype: int64



```
df1=df.drop(["Name", "Sex", "Ticket", "Cabin", "Embarked"],axis=1)
df1.columns

Index(['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare'], dtype='object')
```

```
df1['Age'].isnull().sum()
```



177

```
df1['Age'].fillna(method='ffill')
```



```
<ipython-input-13-ad737854164f>:1: FutureWarning: Series.fillna with 'method' is deprecated and will raise in a future version. Use obj.f
df1['Age'].fillna(method='ffill')
```

Age	
0	22.0
1	38.0
2	26.0
3	35.0
4	35.0
...	...
886	27.0
887	19.0
888	19.0
889	26.0
890	32.0

891 rows × 1 columns

dtype: float64

```
import pandas as pd
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import mutual_info_regression

data=pd.read_csv('/titanic_dataset.csv')
data=data.dropna()
```

```
x=data.drop(['Survived','Name','Ticket'],axis=1)
y=data['Survived']
x
```

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	2	1	female	38.0	1	0	71.2833	C85	C
3	4	1	female	35.0	1	0	53.1000	C123	S
6	7	1	male	54.0	0	0	51.8625	E46	S
10	11	3	female	4.0	1	1	16.7000	G6	S
11	12	1	female	58.0	0	0	26.5500	C103	S
...	...	...	...	...	...	...	...	...	...
871	872	1	female	47.0	1	1	52.5542	D35	S
872	873	1	male	33.0	0	0	5.0000	B51 B53 B55	S
879	880	1	female	56.0	0	1	83.1583	C50	C
887	888	1	female	19.0	0	0	30.0000	B42	S
889	890	1	male	26.0	0	0	30.0000	C148	C

183 rows × 9 columns

Next steps:

Generate code with x

☒ View recommended plots

New interactive sheet

Double-click (or enter) to edit

Suggested code may be subject to a license | BoudhayanBanerjee/political-ad-classifier

```
data["Sex"]=data["Sex"].astype('category')
data["Sex"]=data["Sex"].cat.codes
data["Cabin"]=data["Cabin"].astype('category')
data["Cabin"]=data["Cabin"].cat.codes
data["Embarked"]=data["Embarked"].astype('category')
data["Embarked"]=data["Embarked"].cat.codes
data
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	72	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53,1000	48	2